REMARKS

Claims 1-3, 5, 7-23, 25-27, and 29-31 are pending. Independent claims 1, 20, 30, and 31 have been amended to facilitate prosecution. Claims 4, 6, 24, 28 have been canceled. Other minor amendments have been made to adjust claim references based on claim cancellations. The pending claims including independent claims 1, 20, 30, and 31 were rejected under 35 U.S.C. 102(b) as being anticipated by Sato (USP 6,467,075).

The pending claims were also rejected under 35 U.S.C. 102(e) as being anticipated by Ball (USP 7,203,799). Ball has a common assignee with the present application. A Declaration by the inventor Jeffrey Orion Pritchard has been included indicating that he and Todd Wayne are the ones who actually conceived of the material described on column 9, lines 33-35, "According to various embodiments, the generator program 905 identifies pointers and provides ports for each pointer." Since the cited material was not invented "by another," the Ball reference is believed removed as a 35 U.S.C. 102(e) reference. It should be noted that the Ball patent does not have claims that recite pointers or ports for pointers, and James Ball is believed to be the sole inventor of the Ball patent.

The independent claims have been amended to recite either "providing the hardware acceleration logic with a read port for a pointer read access identified in the portion of the high-level language program" or "providing the hardware acceleration logic with a write port for a pointer write access identified in the portion of the high–level language program."

Sato does describe pointer analysis. "Pointer analysis is a compiler technique to identify at compile-time the potential values of the pointers in the program. This information is used to determine the set of locations the pointer may point to. For synthesis, in the case of loads, stores, and free, this inventors want to synthesize the logic to access, modify or deallocate the location referenced by the pointer. For this purpose, the points-to information must be both safe and accurate: safe because the inventors have to consider all of the locations the pointer may reference and accurate because the smaller the points-to set is, the less logic this inventors have to generate." (column 10, lines 27-38) However, Sato does not provide any hardware acceleration logic with a read port or a write port. Claim 31 further recites "wherein the read port includes a read address line having an address corresponding to the address of the pointer read access." Sato also does not teach or suggest this element.

Sato does mention that pointers may be used to reference any information including all places where information is available, such as wires, ports, registers, memory. "Pointers may be used to reference any variable no matter where its information is available. Pointers must be considered as references: references to memory elements, registers, wires or ports." (column 9, lines 20-23).

However, simply stating that information for pointers may be available on particular wires, ports, registers, memory, does not teach or suggest explicitly creating wires, ports, registers, memory for allowing pointer access. The independent claims explicitly recite creating a hardware acceleration logic write port for a write access or a read port for a read access. Sato does not teach or suggest creating ports on a hardware accelerator for pointer access, but in fact teaches away from providing hardware acceleration logic with a read port or a write port. Sato describes in detail mechanisms for handling pointer access by removing pointer references. For example, independent claim 1 states "A circuit synthesis method of a semiconductor circuit for executing a program with a function of pointers and dynamic allocation, comprising: resolving pointer and dynamic allocation in a code of the program; and changing the code of the program into another code which does not contain said pointer and said dynamic allocation; wherein the semiconductor circuit executes the program with a function of pointers and dynamic allocation when a synthesis of said semiconductor circuit is performed." Sato suggests that the solution is to "change the code of the program into another code which does not contain said pointer and said dynamic allocation."

CONCLUSION

In light of the above remarks relating to independent claims and certain dependent claims, the remaining dependent claims are believed allowable for at least the reasons noted above. Applicants believe that all pending claims are allowable and respectfully request a Notice of Allowance for this application from the Examiner. Should the Examiner believe that a telephone conference would expedite the prosecution of this application, the undersigned can be reached at the telephone number set out below.

Respectfully submitted,
BEYER WEAVER LLP

/Godfrey Audrey Kwan/
Godfrey Audrey Kwan
Reg. No. 46,850

P.O. Box 70250
Oakland, CA 94612-0250
(510) 663-1100